WHAT IS CLAIMED:

1. A method of operating an active debugging environment to debug a virtual application that contains programming language code from a plurality of disparate programming languages, said method comprising:

creating an instance of a script host;

populating said script host with a script text from said virtual application;

interconnecting said virtual application and said script host by way of complementary interfaces therebetween;

dynamically activating said active debugging environment; and

running said script text by said script host under control of said active debugging environment in a manner comprising:

defining at least one event monitoring case by way of a debug user interface;

transparently implementing each of said at least one event monitoring case independent of said programming language code in said script text;

viewing components and characteristics of said virtual application by way of said user interface at any time during debug operations;

identifying an occurrence of each of said at least one event monitoring case; and

activating a predefined response to an occurrence of each of said at least one event monitoring case in positive response to said step of identifying.

2. A method according to claim 1 wherein said step of creating includes:

a first step of establishing a first debug process to catalog and manage each of a plurality of components of each of a plurality of applications within said virtual application; and

a second step of establishing a second debug process to catalog and manage each of said plurality of applications within said virtual application.

3. A method according to claim 1 wherein said step of populating includes:

dynamically establishing a language engine for each programming language represented in said virtual application.

4. A method according to claim 1 wherein said step of interconnecting includes:

activating said script host to expose an object model of each application within said virtual application; and

establishing a standard interface to said script host based upon said object model of each application within said virtual application.

5. A method according to claim 1 wherein said step of defining at least one event monitoring case includes:

imposing at least one debug feature in said script text that alters normal run-time activity of said script text.

6. A method according to claim 1 including:

supporting any one of a plurality of integrated development environment user interfaces to said active debugging environment that is designed by a debug user in a programming language neutral and host independent manner.

7. A method according to claim 6 including:

transmitting any portion of said script text to said integrated development environment user interface to facilitate source code and virtual application context viewing by said debug user; and

receiving data from said integrated development environment user interface indicative of user-input commands to control operation of said active debugging environment.

8. A method according to claim 1 wherein said step of activating a predefined response includes:

generating a view of said script text to a debug user that is relevant to said event; and

dynamically altering said script text by script edit facilities in an integrated development environment user interface.

9. A method according to claim 8 further including:

running said script text on said script host with edited changes absent any run-time interruptions to said active debugging environment.

10. A machine readable memory tangibly embodying instructions executable by a computer to perform a method for debugging at least one application in an active debugging environment wherein said at least one application contains programming language code from multiple programming languages, said method comprising:

a first step of defining a content centric host;

a second step of defining said active debugging environment that is language neutral independent of said multiple programming languages that include compiled programming language code and interpreted programming language code in said application;

generating a virtual application from and aggregate of said at least one application such that said virtual application includes said multiple programming languages and related programming language context; and

executing said virtual application on said content centric host under control of said active debugging environment.

11. A method according to claim 10 wherein said first step of defining a content centric host includes:

establishing a language engine component for each unique programming language associated with said multiple programming languages, said language engine component includes details of programming language specific mapping and debugging features; and

coordinating in-process activities of said content centric host with each of said language engine component and said active debugging environment.

12. A method according to claim 10 wherein said second step of defining said active debugging environment includes:

selecting a language neutral and host neutral user interface and debug feature set for use with said active debugging environment by a human user;

maintaining a real time catalog of programming language components per application present in said virtual application;

maintaining a real time catalog of applications present in said virtual application; and

coordinating language neutral and host neutral communications between said active debugging environment and said content centric host during debug operations.

13. A method according to claim 10 wherein said step of executing includes:

displaying context driven debugging information suitable for human interpretation, where the context driven debugging information is dependent on programming language features supported by a corresponding one of said language engine and intelligence of said host;

first step of maximizing presentable context information for programming language statements run by a smart host; and

second step of maximizing presentable context information for programming language statements run by a dumb host.

14. An active debugging environment to debug at least one application having a mixture of compiled and interpreted programming languages therein, said environment comprising:

first means for defining a content centric host having a language engine specific to each of said compiled and interpreted programming languages;

second means for defining said active debugging environment that is programming language neutral and host neutral;

means for generating a virtual application that includes programming language statements from said compiled and interpreted programming languages and related programming language context; and

means for running said virtual application on said content centric host under control of said active debugging environment; and

means for dynamically debugging said virtual application and any portion of said virtual application that is added to and deleted from said virtual application in response to said means for running.

15. An environment according to claim 14 wherein said content centric host includes:

a language engine component for each unique programming language associated with any one of said compiled and interpreted programming language statements;

means for maintaining a transparent interface of programming language specific structural details and debugging details within each said language engine component; and

a process debug manager to coordinate in-process activities of said content centric host with each of said language engine component and said active debugging environment.

16. An environment according to claim 14 including:

a replaceable generic user interface and debug feature set for use with said active debugging environment by a human user; and

means for maintaining a real time catalog of programming language components present in each of said at least one application of said virtual application;

means for maintaining a real time catalog of each of said at least one application of said virtual application; and

means for coordinating language neutral and host neutral communications between said active debugging environment and said content centric host during debug operations.

17. An environment according to claim 16 that includes:

means for displaying context driven debugging information suitable for human interpretation;

first means for maximizing presentable context information for programming language statements run by a smart host; and

second means for maximizing presentable context information for programming language statements run by a dumb host.